

```
In [5]: import math
```

```
In [11]: import numpy
```

```
In [25]: import pandas as pd  
file_path = r"C:\Users\LENOVO\Desktop\owid-covid-data.csv"  
df = pd.read_csv(file_path)  
print(df.head())
```

	iso_code	continent	location	date	total_cases	new_cases	\
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	

	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	\
0	NaN	0.0	0.0	NaN	...	
1	NaN	0.0	0.0	NaN	...	
2	NaN	0.0	0.0	NaN	...	
3	NaN	0.0	0.0	NaN	...	
4	NaN	0.0	0.0	NaN	...	

	male_smokers	handwashing_facilities	hospital_beds_per_thousand	\
0	NaN	37.746	0.5	
1	NaN	37.746	0.5	
2	NaN	37.746	0.5	
3	NaN	37.746	0.5	
4	NaN	37.746	0.5	

	life_expectancy	human_development_index	population	\
0	64.83	0.511	41128772	
1	64.83	0.511	41128772	
2	64.83	0.511	41128772	
3	64.83	0.511	41128772	
4	64.83	0.511	41128772	

	excess_mortality_cumulative_absolute	excess_mortality_cumulative	\
0	NaN	NaN	
1	NaN	NaN	
2	NaN	NaN	
3	NaN	NaN	
4	NaN	NaN	

	excess_mortality	excess_mortality_cumulative_per_million
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

[5 rows x 67 columns]

```
In [37]: # Dimensioni del dataset  
print("Dimensioni del dataset:", data.shape)
```

Dimensioni del dataset: (429435, 67)

```
In [39]: # Metadati  
print("Informazioni sul dataset:")  
print(data.info())
```

Informazioni sul dataset:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 429435 entries, 0 to 429434

Data columns (total 67 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	iso_code	429435 non-null	object
1	continent	402910 non-null	object
2	location	429435 non-null	object
3	date	429435 non-null	object
4	total_cases	411804 non-null	float64
5	new_cases	410159 non-null	float64
6	new_cases_smoothed	408929 non-null	float64
7	total_deaths	411804 non-null	float64
8	new_deaths	410608 non-null	float64
9	new_deaths_smoothed	409378 non-null	float64
10	total_cases_per_million	411804 non-null	float64
11	new_cases_per_million	410159 non-null	float64
12	new_cases_smoothed_per_million	408929 non-null	float64
13	total_deaths_per_million	411804 non-null	float64
14	new_deaths_per_million	410608 non-null	float64
15	new_deaths_smoothed_per_million	409378 non-null	float64
16	reproduction_rate	184817 non-null	float64
17	icu_patients	39116 non-null	float64
18	icu_patients_per_million	39116 non-null	float64
19	hosp_patients	40656 non-null	float64
20	hosp_patients_per_million	40656 non-null	float64
21	weekly_icu_admissions	10993 non-null	float64
22	weekly_icu_admissions_per_million	10993 non-null	float64
23	weekly_hosp_admissions	24497 non-null	float64
24	weekly_hosp_admissions_per_million	24497 non-null	float64
25	total_tests	79387 non-null	float64
26	new_tests	75403 non-null	float64
27	total_tests_per_thousand	79387 non-null	float64
28	new_tests_per_thousand	75403 non-null	float64
29	new_tests_smoothed	103965 non-null	float64
30	new_tests_smoothed_per_thousand	103965 non-null	float64
31	positive_rate	95927 non-null	float64
32	tests_per_case	94348 non-null	float64
33	tests_units	106788 non-null	object
34	total_vaccinations	85417 non-null	float64

```

35 people_vaccinated      81132 non-null float64
36 people_fully_vaccinated 78061 non-null float64
37 total_boosters         53600 non-null float64
38 new_vaccinations       70971 non-null float64
39 new_vaccinations_smoothed 195029 non-null float64
40 total_vaccinations_per_hundred 85417 non-null float64
41 people_vaccinated_per_hundred 81132 non-null float64
42 people_fully_vaccinated_per_hundred 78061 non-null float64
43 total_boosters_per_hundred 53600 non-null float64
44 new_vaccinations_smoothed_per_million 195029 non-null float64
45 new_people_vaccinated_smoothed 192177 non-null float64
46 new_people_vaccinated_smoothed_per_hundred 192177 non-null float64
47 stringency_index       196190 non-null float64
48 population_density     360492 non-null float64
49 median_age             334663 non-null float64
50 aged_65_older          323270 non-null float64
51 aged_70_older          331315 non-null float64
52 gdp_per_capita          328292 non-null float64
53 extreme_poverty         211996 non-null float64
54 cardiovasc_death_rate  328865 non-null float64
55 diabetes_prevalence     345911 non-null float64
56 female_smokers          247165 non-null float64
57 male_smokers            243817 non-null float64
58 handwashing_facilities 161741 non-null float64
59 hospital_beds_per_thousand 290689 non-null float64
60 life_expectancy         390299 non-null float64
61 human_development_index 319127 non-null float64
62 population             429435 non-null int64
63 excess_mortality_cumulative_absolute 13411 non-null float64
64 excess_mortality_cumulative 13411 non-null float64
65 excess_mortality        13411 non-null float64
66 excess_mortality_cumulative_per_million 13411 non-null float64
dtypes: float64(61), int64(1), object(5)
memory usage: 219.5+ MB
None

```

```

In [41]: # Rimuovere righe con valori nulli nei continenti
data = data.dropna(subset=["continent"])

```

```
In [43]: # Calcolare i casi totali per continente
cases_per_continent = data.groupby("continent")["total_cases"].max()
print("Casi totali per continente:")
print(cases_per_continent)
```

```
Casi totali per continente:
continent
Africa          4072765.0
Asia            99373219.0
Europe          38997490.0
North America   103436829.0
Oceania         11861161.0
South America   37511921.0
Name: total_cases, dtype: float64
```

```
In [45]: # Casi totali globali
global_cases = cases_per_continent.sum()
```

```
In [47]: print(data.columns)
print(data["total_cases"].isnull().sum()) # Mostra quanti valori mancanti ci sono
```

```
Index(['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
      'new_cases_smoothed', 'total_deaths', 'new_deaths',
      'new_deaths_smoothed', 'total_cases_per_million',
      'new_cases_per_million', 'new_cases_smoothed_per_million',
      'total_deaths_per_million', 'new_deaths_per_million',
      'new_deaths_smoothed_per_million', 'reproduction_rate', 'icu_patients',
      'icu_patients_per_million', 'hosp_patients',
      'hosp_patients_per_million', 'weekly_icu_admissions',
      'weekly_icu_admissions_per_million', 'weekly_hosp_admissions',
      'weekly_hosp_admissions_per_million', 'total_tests', 'new_tests',
      'total_tests_per_thousand', 'new_tests_per_thousand',
      'new_tests_smoothed', 'new_tests_smoothed_per_thousand',
      'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations',
      'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
      'new_vaccinations', 'new_vaccinations_smoothed',
      'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
      'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
      'new_vaccinations_smoothed_per_million',
      'new_people_vaccinated_smoothed',
      'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
      'population_density', 'median_age', 'aged_65_older', 'aged_70_older',
      'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate',
      'diabetes_prevalence', 'female_smokers', 'male_smokers',
      'handwashing_facilities', 'hospital_beds_per_thousand',
      'life_expectancy', 'human_development_index', 'population',
      'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
      'excess_mortality', 'excess_mortality_cumulative_per_million'],
      dtype='object')
```

11194

```
In [49]: # Casi totali globali
        global_cases = cases_per_continent.sum()

        # Mostra il risultato
        print("Casi totali globali:", global_cases)
```

Casi totali globali: 295253385.0

```
In [51]: # Percentuali per continente
        cases_percentage = (cases_per_continent / global_cases) * 100
```

```
print("Percentuale dei casi per continente:")
print(cases_percentage)
```

Percentuale dei casi per continente:

continent

Africa 1.379413

Asia 33.656928

Europe 13.208143

North America 35.033241

Oceania 4.017282

South America 12.704993

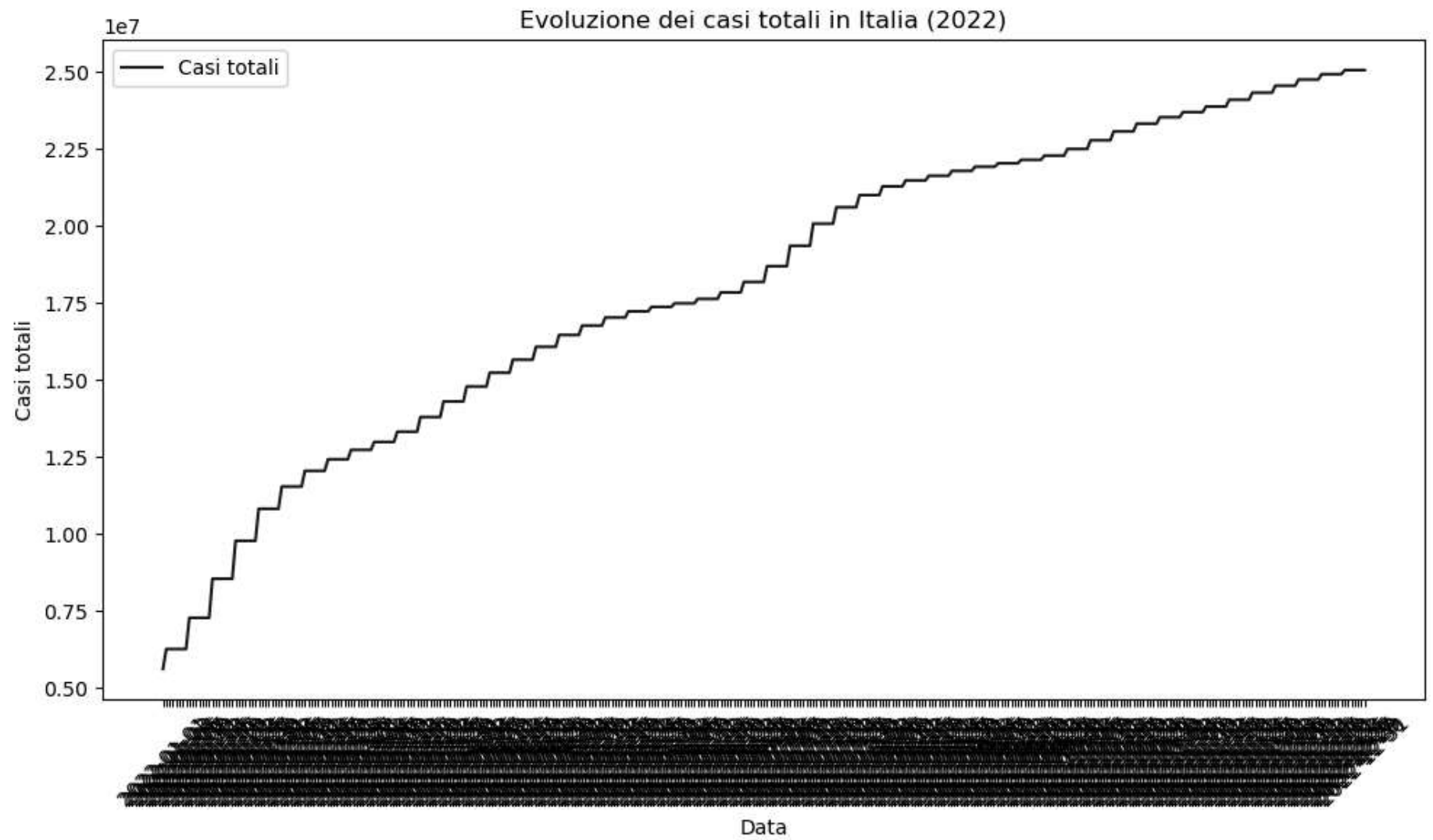
Name: total\_cases, dtype: float64

```
In [53]: import matplotlib.pyplot as plt
```

```
In [57]: # Filtrare i dati per l'Italia e per l'anno 2022
italy_data = data[(data["location"] == "Italy") & (data["date"].str.startswith("2022"))]

# Grafico casi totali
plt.figure(figsize=(10, 6))
plt.plot(italy_data["date"], italy_data["total_cases"], label="Casi totali", color="blue")
plt.title("Evoluzione dei casi totali in Italia (2022)")
plt.xlabel("Data")
plt.ylabel("Casi totali")
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

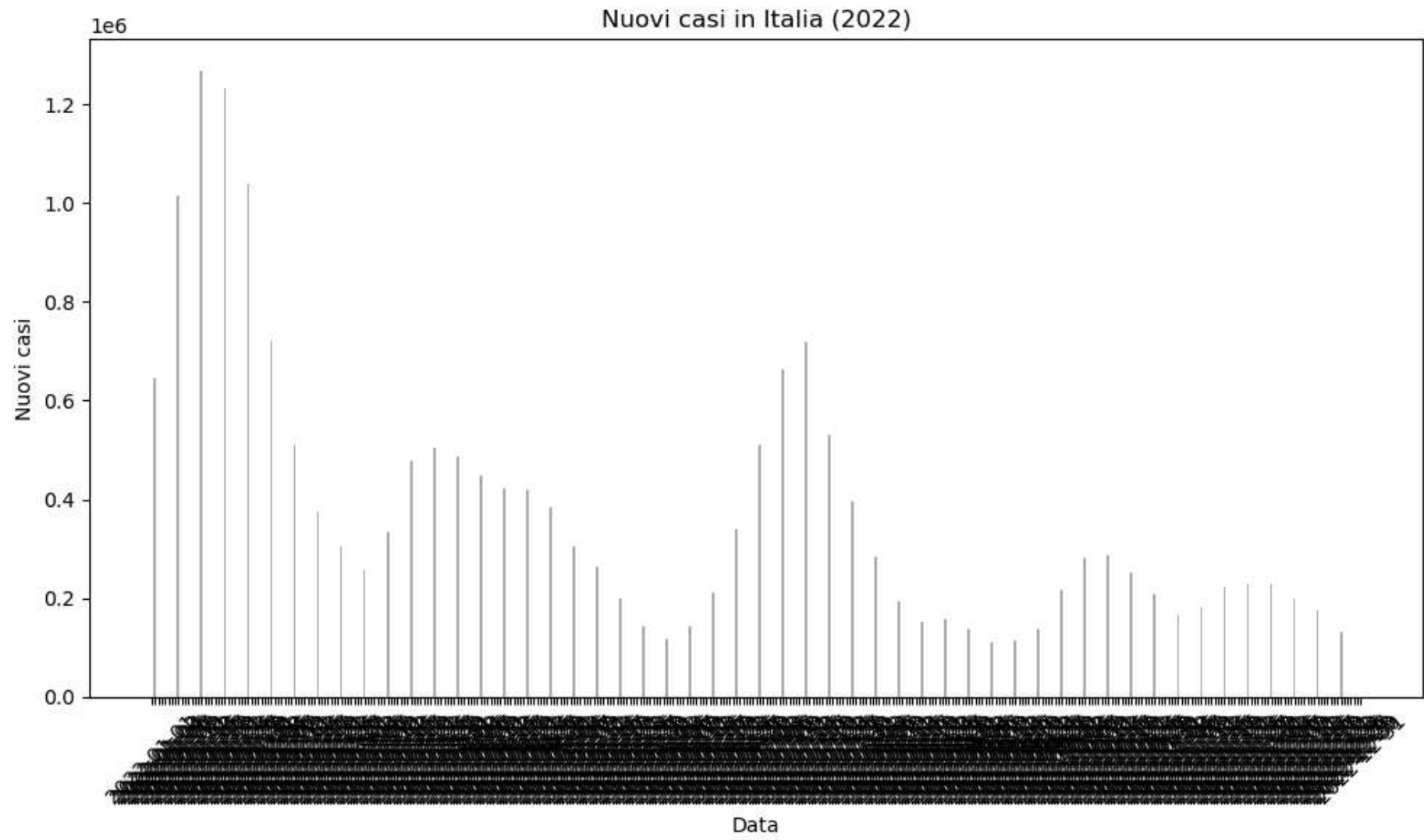




```
In [59]: # Filtrare dati con nuovi casi settimanali validi
weekly_data = italy_data.dropna(subset=["new_cases"])

# Grafico nuovi casi
plt.figure(figsize=(10, 6))
plt.bar(weekly_data["date"], weekly_data["new_cases"], color="orange")
plt.title("Nuovi casi in Italia (2022)")
```

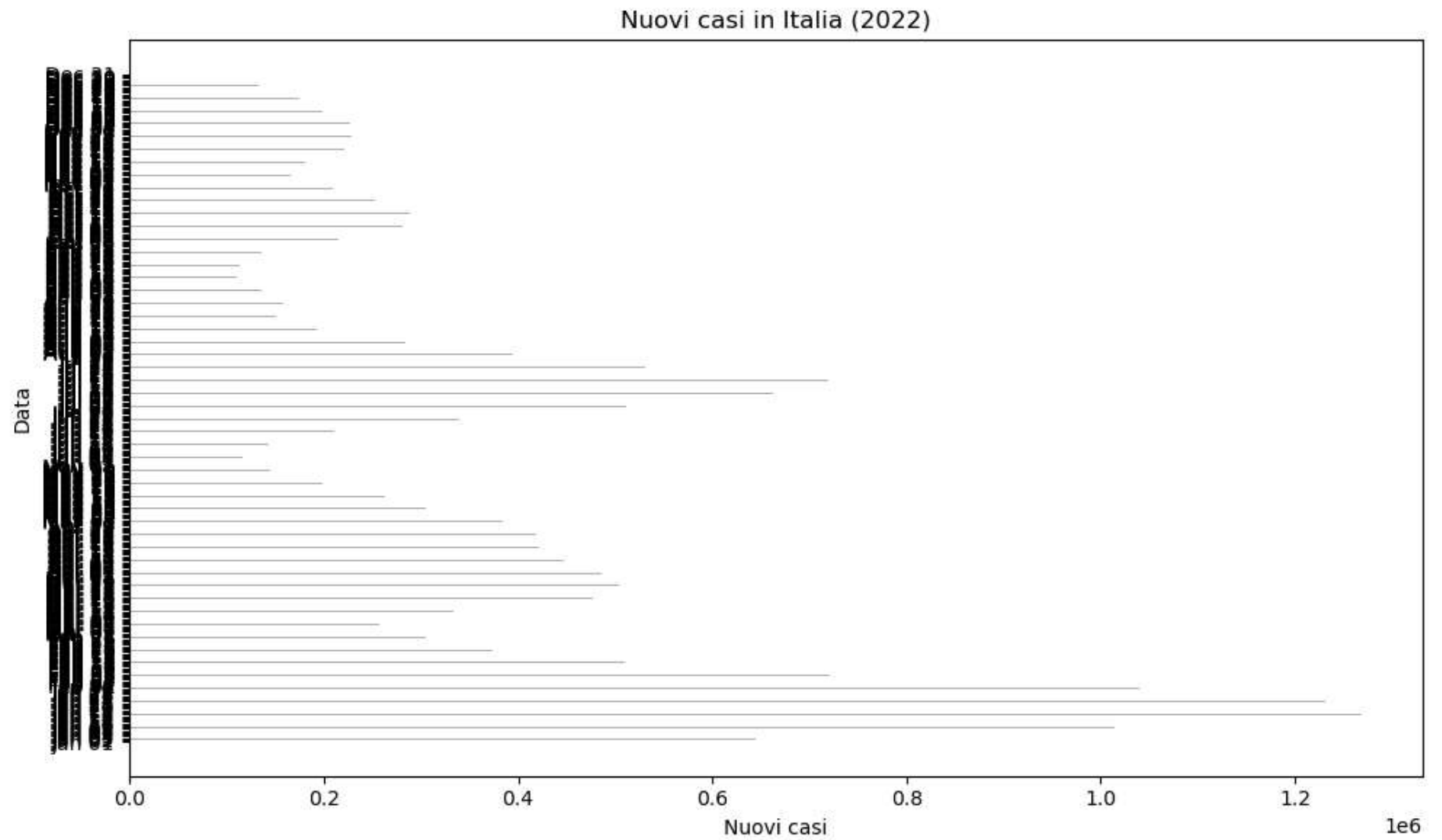
```
plt.xlabel("Data")
plt.ylabel("Nuovi casi")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
In [65]: plt.figure(figsize=(10, 6))
plt.barh(weekly_data["date"], weekly_data["new_cases"], color="orange") # Usa barh per barre orizzontali
```

```
plt.title("Nuovi casi in Italia (2022)")
plt.ylabel("Data")
plt.xlabel("Nuovi casi")

# Imposta il formato delle date sull'asse y
plt.gca().yaxis.set_major_formatter(mdates.DateFormatter("%b %d"))
plt.tight_layout()
plt.show()
```

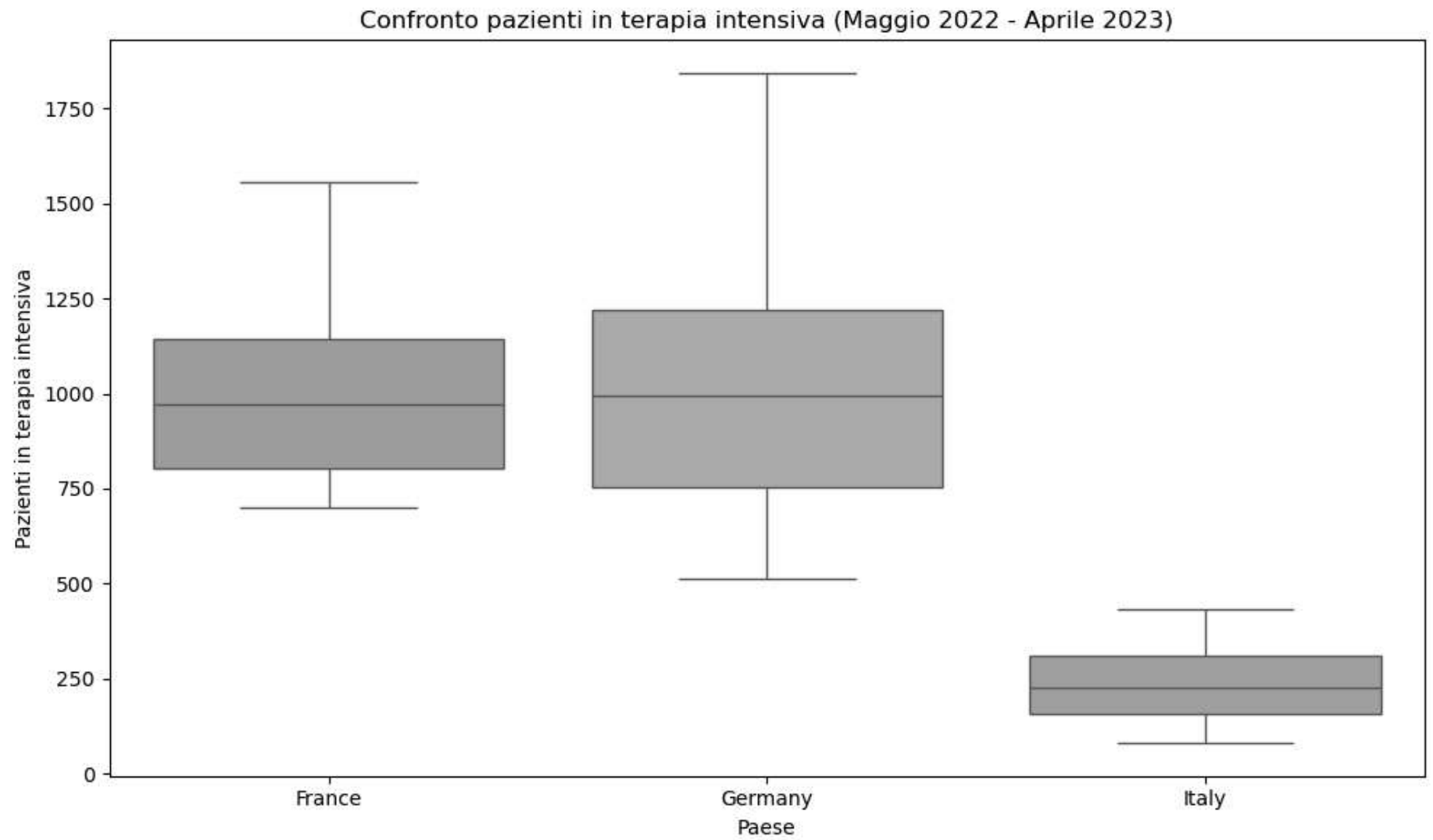


```
In [69]: # Filtrare dati per le nazioni e il periodo
icu_countries = data[
    (data["location"].isin(["Italy", "Germany", "France"])) &
    (data["date"] >= "2022-05-01") & (data["date"] <= "2023-04-30")
]

# Creare il boxplot
```

```
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.boxplot(x="location", y="icu_patients", hue="location", data=icu_countries, palette="Set2", legend=False)
plt.title("Confronto pazienti in terapia intensiva (Maggio 2022 - Aprile 2023)")
plt.xlabel("Paese")
plt.ylabel("Pazienti in terapia intensiva")
plt.tight_layout()
plt.show()
```



```
In [71]: # Filtrare i dati per il 2023 e i paesi richiesti
hospital_data = data[
    (data["location"].isin(["Italy", "Germany", "France", "Spain"])) &
    (data["date"].str.startswith("2023"))
]

# Calcolare la somma dei pazienti ospedalizzati
```

```
hospitalized_sum = hospital_data.groupby("location")["hosp_patients"].sum()
print("Pazienti ospedalizzati nel 2023 per paese:")
print(hospitalized_sum)
```

Pazienti ospedalizzati nel 2023 per paese:

location

France 1382574.0

Germany 0.0

Italy 1175272.0

Spain 354602.0

Name: hosp\_patients, dtype: float64

```
In [73]: # Verificare la presenza di dati nulli
null_counts = hospital_data["hosp_patients"].isnull().sum()
print("Numero di valori nulli nei dati di pazienti ospedalizzati:", null_counts)

# Commento sulla gestione:
if null_counts > 0:
    print("I valori nulli possono essere sostituiti, ad esempio, con la media settimanale o mensile.")
else:
    print("Non ci sono valori nulli da gestire.")
```

Numero di valori nulli nei dati di pazienti ospedalizzati: 843

I valori nulli possono essere sostituiti, ad esempio, con la media settimanale o mensile.

In [ ]: